NWS Praktikumsaufgaben

Grundlegende Bedienelemente des Internet-Analyse-Systems

Stand: 2011/11/15 -9-

Dominique Petersen <petersen@internet-sicherheit.de>
Institut für Internet-Sicherheit, https://www.internet-sicherheit.de
Fachhochschule Gelsenkirchen

Inhaltsverzeichnis

Einleitung

Drei-Wege-Handshake[1]

TCP Netzwerkverbindungen werden durch den „TCP Handshake“ initiiert und durch den „TCP Teardown“ abgebaut. Diese Vorgänge werden durch die TCP Flags im TCP Header gesteuert.

Der Drei-Wege-Handshake ist die Bezeichnung für ein Verfahren, um eine in Bezug auf Übertragungsverluste sichere Datenübertragung zwischen zwei Instanzen zu ermöglichen. Obwohl überwiegend in der Netzwerktechnik verwendet, ist der Drei-Wege-Handshake nicht auf diese beschränkt.

Verbindungsaufbau

Beim Aufbau einer TCP-Verbindung kommt der Drei-Wege-Handshake zum Einsatz. Der Rechner, der die Verbindung aufbauen will, sendet dem anderen ein *SYN*-Paket (von engl. *synchronize*) mit einer Sequenznummer *x*. Die Sequenznummern sind dabei für die Sicherstellung einer vollständigen Übertragung in der richtigen Reihenfolge und ohne Duplikate wichtig. Es handelt sich also um ein Paket, dessen *SYN-Bit* im Paketkopf gesetzt ist (siehe TCP-Header). Die Start-Sequenznummer ist eine beliebige Zahl, deren Generierung von der jeweiligen TCP-Implementierung abhängig ist. Sie sollte jedoch möglichst zufällig sein, um Sicherheitsrisiken zu vermeiden.

Die Gegenstelle (siehe Skizze) empfängt das Paket. Ist der Port geschlossen, antwortet sie mit einem TCP-RST um zu signalisieren, dass keine Verbindung aufgebaut werden kann. Ist der Port geöffnet, sendet sie in einem eigenen SYN-Paket im Gegenzug ihre Start-Sequenznummer *y* (die ebenfalls beliebig und unabhängig von der Start-Sequenznummer der Gegenstelle ist). Zugleich bestätigt sie den Erhalt des ersten SYN-Pakets, indem sie die Sequenznummer *x* um eins erhöht und im ACK-Teil (von engl. *acknowledgment* = *Bestätigung*) des Headers zurückschickt.

Der Client bestätigt zuletzt den Erhalt des SYN/ACK-Pakets durch das Senden eines eigenen ACK-Pakets mit der Sequenznummer *y+1*. Dieser Vorgang wird auch als „Forward Acknowledgement“ bezeichnet. Außerdem sendet der Client den Wert *x+1* aus Sicherheitsgründen ebenso zurück.

1 http://de.wikipedia.org/wiki/Transmission_Control_Protocol#Der_Drei-Wege-Handschlag

Dieses ACK-Segment erhält der Server, das ACK-Segment ist durch das gesetzte ACK-Flag gekennzeichnet. Die Verbindung ist damit aufgebaut.

1.	SYN-SENT	→	<SEQ=100><CTL=SYN>	→	SYN-RECEIVED
2.	SYN/ACK-RECEIVED	←	<SEQ=300><ACK=101><CTL=SYN,ACK>	←	SYN/ACK-SENT
3.	ACK-SENT	→	<SEQ=101><ACK=301><CTL=ACK>	→	ESTABLISHED

Verbindungsabbau

Der geregelte Verbindungsabbau erfolgt ähnlich. Statt des SYN-Bits kommt das FIN-Bit (von engl. *finish* = *Ende, Abschluss*) zum Einsatz, welches anzeigt, dass keine Daten mehr vom Sender kommen. Der Erhalt des Pakets wird wiederum mittels ACK bestätigt. Der Empfänger des FIN-Pakets sendet zuletzt seinerseits ein FIN-Paket, das ihm ebenfalls bestätigt wird.

Obwohl eigentlich vier Wege genutzt werden, handelt es sich beim Verbindungsabbau auch um einen Drei-Wege-Handshake, da die ACK- und FIN-Operationen vom Server zum Client als ein Weg gewertet werden. Zudem ist ein verkürztes Verfahren möglich, bei dem FIN und ACK genau wie beim Verbindungsaufbau im selben Paket untergebracht werden. Die *maximum segment lifetime* (MSL) ist die maximale Zeit, die ein Segment im Netzwerk verbringen kann, bevor es verworfen wird. Nach dem Senden des letzten ACKs wechselt der Client in einen zwei MSL andauernden Wartezustand (Waitstate), in dem alle verspäteten Segmente verworfen werden. Dadurch wird sichergestellt, dass keine verspäteten Segmente als Teil einer neuen Verbindung fehlinterpretiert werden. Außerdem wird eine korrekte Verbindungsterminierung sichergestellt. Geht ACK *y+1* verloren, läuft beim Server der Timer ab, und das LAST_ACK Segment wird erneut übertragen.

Control-Flags

Control-Flags sind zweiwertige Variablen, mit den möglichen Zuständen *gesetzt* und *nicht gesetzt*, welche zur Kennzeichnung bestimmter für die Kommunikation und Weiterverarbeitung der Daten wichtiger Zustände benötigt werden. Im folgenden werden die Flags des TCP-Headers und die von ihrem Zustand abhängigen, auszuführenden Aktionen beschrieben.

URG

Ist das Urgent-Flag *(urgent = dringend)* gesetzt, so werden die Daten, auf die das *Urgent-Pointer*-Feld zeigt, sofort von der Anwendung bearbeitet. Dabei unterbricht die Anwendung die Verarbeitung der Daten des aktuellen TCP-Segments und liest das Byte aus, auf das der Urgent-Pointer zeigt. Dieses Verfahren ist fern verwandt mit einem Softwareinterrupt. Dieses Flag kann zum Beispiel verwendet werden, um eine Anwendung auf dem Empfänger abzubrechen. Das Verfahren wird nur äußerst selten benutzt, Beispiele sind rlogin und telnet.

ACK

Das *Acknowledgment*-Flag hat in Verbindung mit der *Acknowledgment*-Nummer zwei unterschiedliche Aufgaben. Zum einen dient es bei gleichzeitig gesetztem SYN-Flag zur Bestätigung beim Drei-Wege-Handshake, zum anderen wird es ohne SYN-Flag zur Bestätigung von TCP-Segmenten beim Datentransfer genutzt. Die *Acknowledgment*-Nummer ist nicht gültig, wenn das Flag nicht gesetzt ist.

PSH

Das *Push*-Flag hat die Aufgabe, die Daten unter Umgehung des Puffers, eines Speichers für die Zwischenlagerung von Daten, sofort an die Anwendung weiterzuleiten. Hilfreich ist dies, wenn man zum Beispiel bei einer Telnet-Sitzung einen Befehl an den Empfänger senden will. Würde dieser Befehl erst im Puffer zwischengespeichert werden, so würde dieser (stark) verzögert abgearbeitet werden.

RST

Das *Reset*-Flag wird verwendet, wenn eine Verbindung abgebrochen werden soll. Dies geschieht zum Beispiel bei technischen Problemen oder zur Abweisung unerwünschter Verbindungen.

SYN

Pakete mit gesetztem SYN-Flag initiieren eine Verbindung, d.h. beginnen den Drei-Wege-Handshake. Der Server antwortet normalerweise entweder mit SYN+ACK, wenn er bereit ist, die Verbindung anzunehmen, andernfalls mit RST. Dient der Synchronisation von *Sequenznummern* beim Verbindungsaufbau (daher die Bezeichnung SYN).

FIN

Dieses Finish-Flag dient zur Freigabe der Verbindung und zeigt an, dass keine Daten mehr vom Sender kommen. Die FIN- und SYN-Flags haben Sequenznummern, damit diese in der richtigen Reihenfolge abgearbeitet werden.

Verhältnis der Flag-Kombinationen

Über das Verhältnis der Anzahl der Flag-Kombinationen können Informationen über das analysierte Netzwerk erlangt werden. Relevante Flag-Kombinationen sind SYN, SYN/ACK, FIN und FIN/ACK.

Beispiel (fiktiv): In einem Zeitraum von 5 Minuten konnte beobachtet werden, das über eine Kommunikationsleitung folgende Pakete mit oben angegebenen Flag-Kombinationen übertragen wurden:

SYN: 112 Pakete
SYN/ACK: 6 Pakete
FIN: 60 Pakete
FIN/ACK: 22 Pakete

Daher ergibt sich das folgende Verhältnis der Pakete untereinander:

SYN: 56 %
SYN/ACK: 3 %
FIN: 30 %
FIN/ACK: 11%

SYN-Scan[1]

Hacker verwenden für einen sog. SYN-Scan eine Technik, die sich der TCP-Flags bedient.

1 http://de.wikipedia.org/wiki/Portscanner#TCP_SYN_Scan

Beim TCP SYN Scan wird ein TCP-Paket mit SYN-Flag an den Ziel-Host gesendet, um einen Verbindungsversuch vorzutäuschen. Die Antwort des Hosts gibt Aufschluss über den Port: Sendet er ein SYN/ACK-Paket, den zweiten Teil des Drei-Wege-Handshakes von TCP, akzeptiert der Port Verbindungen und ist daher offen. Der Quell-Host antwortet dann in der Regel mit einem RST-Paket, um die Verbindung wieder abzubauen (dies geschieht meist allerdings nicht durch den Portscanner, sondern durch das Betriebssystem, da offiziell kein Verbindungsversuch unternommen wurde). Sendet der Host ein RST-Paket, ist der Port geschlossen. Sendet der Ziel-Host überhaupt kein Paket, ist ein Paketfilter vorgeschaltet.

Der Vorteil dieser Methode ist, dass die gescannte Anwendung keinen Verbindungsversuch erkennt. Deshalb erscheint die Verbindung nicht in den Logdateien und kann daher auch nicht analysiert werden. Jede bessere Firewall erkennt diesen Scan allerdings. Auf den meisten Quell-Systemen sind außerdem Systemverwalterrechte notwendig, weil TCP-Pakete vom Portscanner handgefertigt werden müssen.

TCP SYN Scans lassen sich für Denial-of-Service-Attacken in Form von SYN-Flood nutzen.

Aufgabenstellungen

Hinweis: Die Praktikumsaufgaben dürfen ausschließlich mit den Sonden 8000001 (Inbound FB5), 8000002 (Outbound FB5) und 9000001 (DMAG) bearbeitet werden. Die Beobachtungen und Analysen müssen – soweit nicht anders mit der Praktikumsbetreuung abgesprochen – im Zeitraum vom 01.02.2011 bis zum aktuellen Datum durchgeführt werden. Für die Versuche ausgenommen sind fehlende Messbereiche, wie beispielsweise vom 28.10. bis zum 11.11.2011 im FB5.

Aufgabe 1.)

Welches Verhältnis der Pakete mit ausschließlich gesetztem SYN Flag, ausschließlich gesetztem FIN Flag, gesetztem SYN und gesetztem ACK Flag sowie gesetztem FIN und gesetztem ACK Flag ist theoretisch immer zu erwarten? Warum?

___ % SYN ___ % SYN/ACK ___% FIN ___ % FIN/ACK

Aufgabe 2.)

Welches Verhältnis erwarten sie ungefähr in der Praxis? Warum?

___ % SYN ___ % SYN/ACK ___% FIN ___ % FIN/ACK

Aufgabe 3.)

Wieso können Pakete mit ausschließlich gesetztem ACK Flag nicht für die Analyse des TCP-Handshake verwendet werden, obwohl der Handshake mit so einem Paket abgeschlossen wird (schlüssige Begründung)?

Aufgabe 4.)

Welches Verhältnis liegt auf dem Backbone des FB5 vor? Analysieren sie die beiden Sonden des FB5 im Verbund über einen Zeitraum von mindestens ca. 4 Wochen. Belegen sie ihr Ergebnis mit Screenshots.

Weicht das Ergebnis von der Erwartung ab? Wenn ja, wieso?

___ % SYN ___ % SYN/ACK ___% FIN ___ % FIN/ACK

Aufgabe 5.)

Suchen sie in den Daten des FB5 nach einem SYN-Scan-Angriff. Dokumentieren Sie Ihren Fund mit Screenshots in unterschiedlichen Ansichten. Beschreiben sie den Angriff. Nehmen sie den Deskriptor für Pakete mit gesetztem RST-Flag in die Analyse auf und beschreiben sie die Zusammenhänge.

Aufgabe 6.)

Welche anderen Schad-Szenarien kann man mit dem IAS ihrer Meinung nach erkennen? Denken Sie sowohl über Viren, Würmer und Trojanische Pferde nach, als auch über andere Schadensarten, wie beispielsweise SPAM, Malware oder Angiffe z.B. mit/gegen SIP/RTP, SMTP, HTTP, IRC, etc.

Versuchen Sie, wenigstens ein weiteres Schad-Szenario in den Daten des FB5-Backbone zu finden. Beschreiben Sie das gefundene Szenario detailliert (sowohl in der Theorie des Szenarios

selbst, als auch in seiner gefundenen Ausprägung) und belegen Sie Ihren Fund mit Screenshots.

Aufgabe 7.)

Überlegen Sie sich, wie das Internet-Analyse-System noch weiter verbessert werden kann. Dies betrifft u.A. die Benutzung und Bedienbarkeit des EagleX-Clients sowie die Fähigkeiten, Angriffe bzw. Anomalien analysieren zu können (weitere Deskriptoren, Protokolle, Plugins, ...). Welche Funktionalitäten wünschen Sie sich, um das Internet-Analyse-System weiter zu optimieren?

Allgemeines

Die Aufgaben sind **schriftlich ausgearbeitet bis zum 15.01.2012 im PDF-Format per E-Mail** an mich (petersen@internet-sicherheit.de) abzugeben. Zu dem beinhalteten **Deckblatt** gehören selbstverständlich die üblichen Informationen (Gruppennummer, Namen etc.). Der erwartete Umfang der Ausarbeitung, **exklusive** der Screenshots, sieht so aus:

Aufgaben	Erwartete Seiten
1., 2., 3.	1
4.	1
5.	1
6.	1++
7.	1++

Beachten Sie, dass die Ausarbeitungen anteilig in die Bewertung und damit in die Endnote im Fach NWS mit einfließen kann.